

## **1. Introduction:**

Encryption of data on images provides a safe and secure transmission of data between the sending and receiving party. The data/text which the sender wants to transmit is selected first and then an image is chosen. The chosen text is then encrypted in the image such that the data is not visible to any third party. After encryption is performed, the image is sent to receiving party and the receiver decrypts the data. This decryption process can only be performed by an authenticated receiver using this application.

### **1.1. Problem Definition**

With the rise of technology, there has also been rise in cybercrime. Many hacker are trying to steal sensitive data such as bank account details, private business messages, etc. To create an application which can transmit message in form of disguise that can only be accessed by the receiver.

### **1.2. Objectives**

Main objective of this application is to provide a secure and secret transmission of text by encrypting it on an image and which can only be decrypted by an authenticated receiver on the same application

### **1.3. Literature Survey:**

Our Project uses steganography to encrypt and decrypt the image. Steganography consist of various methodologies types such as

1. Least significant bit insertion (LSB)
2. Masking and filtering
3. Transform technique

Our project uses masking and filtering method to encrypt and decrypt the text on image. The masking and filtering method is usually restricted to 24 bit images. This method uses specific masking algorithms or mathematical formulae to select specific pixels from the cover image. The chosen pixels can be used to embed the secret information after due pre-processing. Thus the embedded secret information looks as an integral part of the cover image. By using proper filter (mathematical expression) to select the pixels, the security of the image stenographic method is enhanced and the embedding capacity can be increased by choosing many pixels.

# 1. Software Requirement Specification

## 2.1. Functional Requirements

### ❖ The user can perform following operations:

- Create/deletion of his own account.
- Change Password
- Send and receive messages

### ❖ The user cannot perform following operations:

- Can't view unauthorized message of other
- Can't send message to the user who are not registered to the system.
- User can't send message to himself.

## 2.2. Non-functional Requirements

### ❖ Performance Requirements:

- ✓ User can send the message within 10 seconds with average internet speed.
- ✓ The message consist of real-time listener that it is synchronized.
- ✓ The message can be sent and received simultaneously,

### ❖ Security Requirements:

- ✓ Only the authenticated user can access the system.
- ✓ The message can only be accessed by authenticated person only in form of image which can be later be decrypted.

## 2.3. Hardware and Software Requirements:

### ❖ Recommended Operating Requirements:

Operating System : Android 5.0 and above

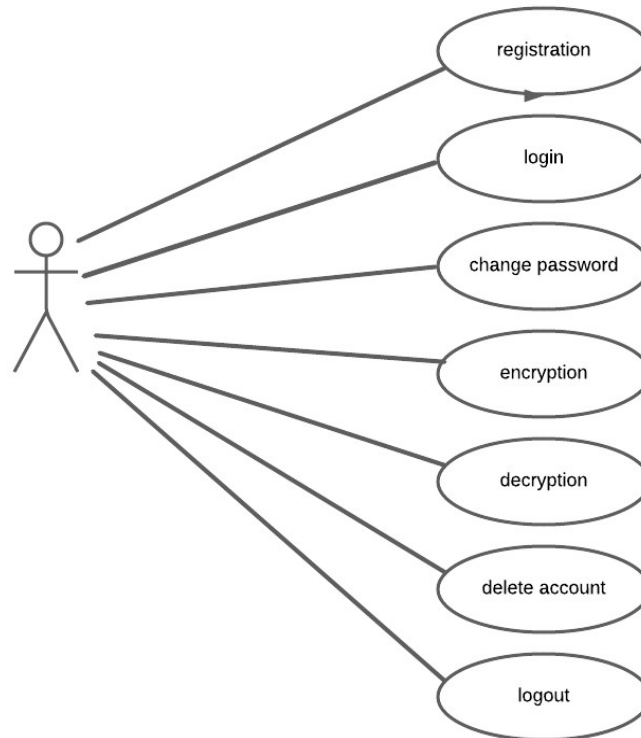
### ❖ Hardware Requirements:

Ram : At least 512 MB of RAM

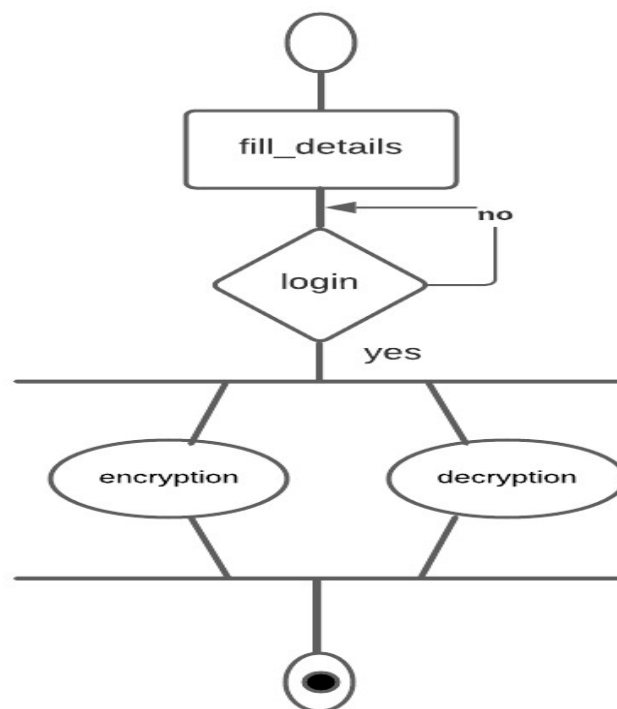
Memory : At least 100 MB of storage

## 2. System Design

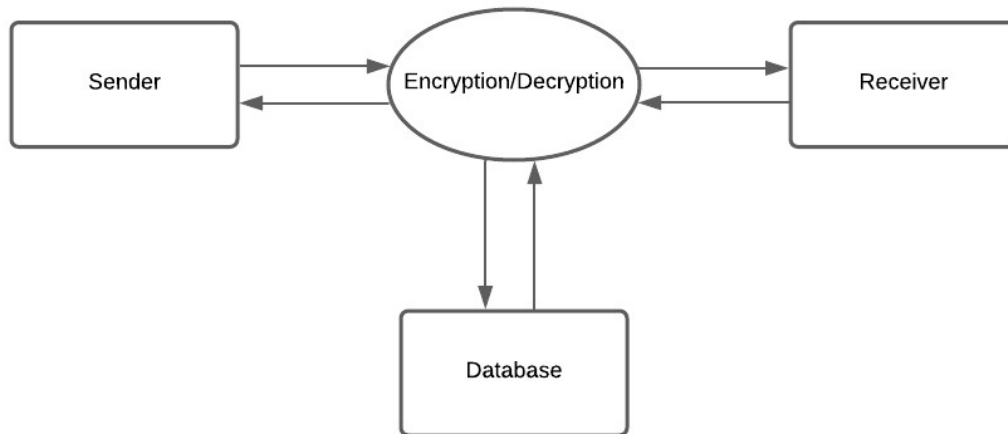
- Use Case:



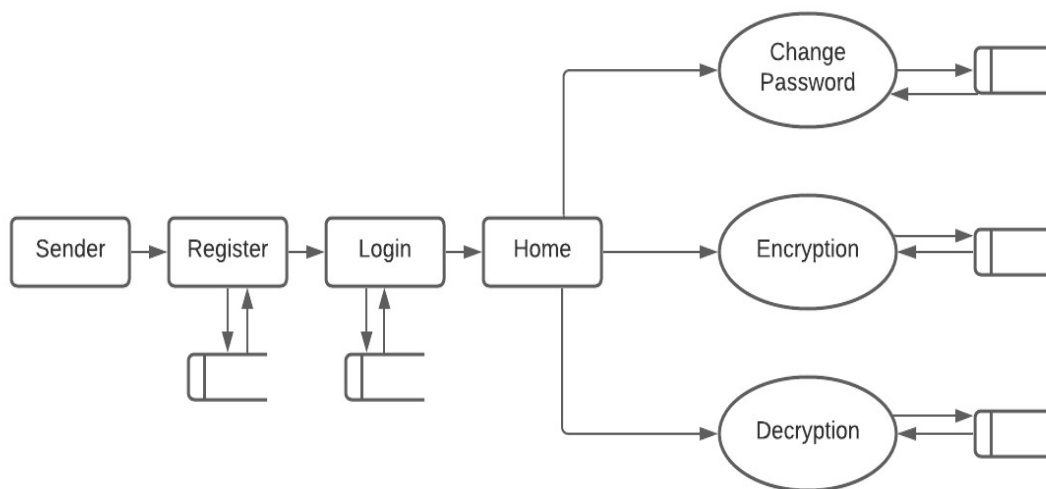
- Activity Diagram



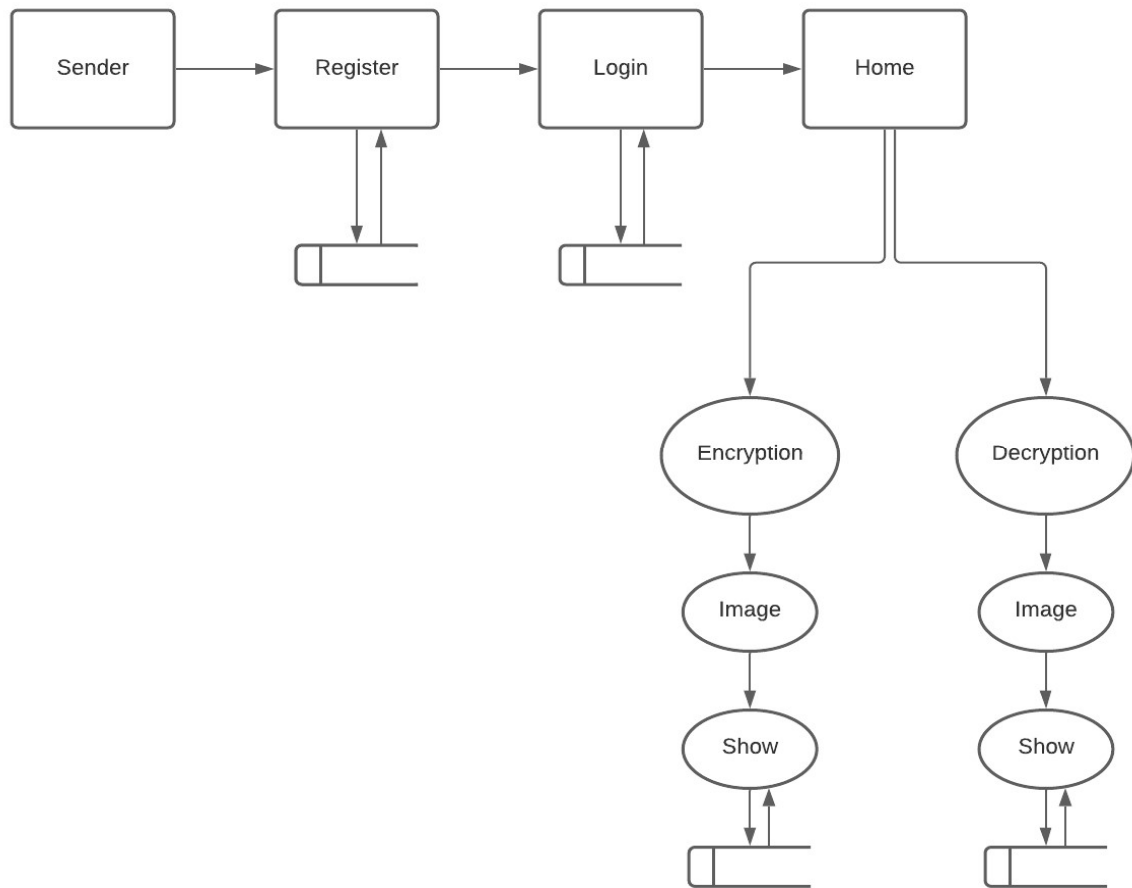
- **Data Flow Diagram:**



**LEVEL 0**



**LEVEL 1**



## LEVEL 2

### 3. System Implementation

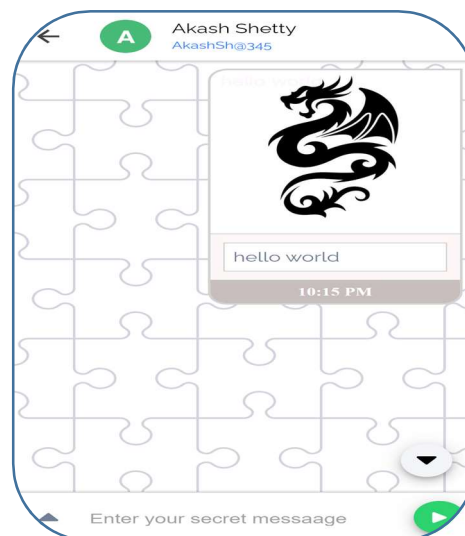
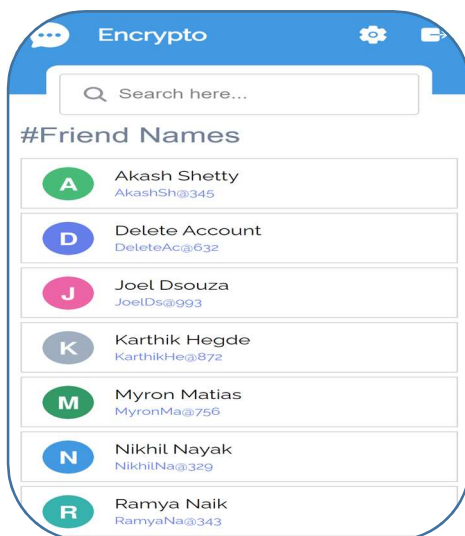
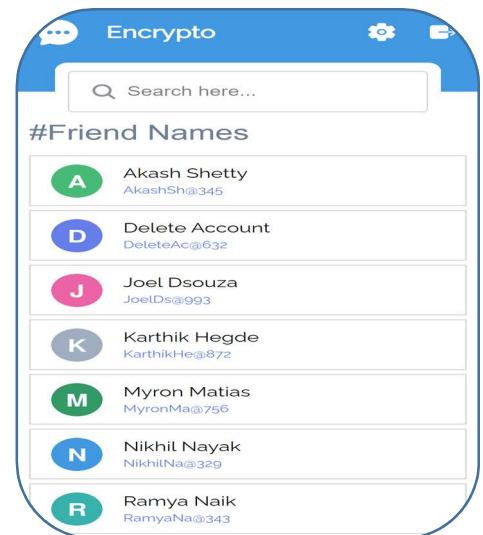
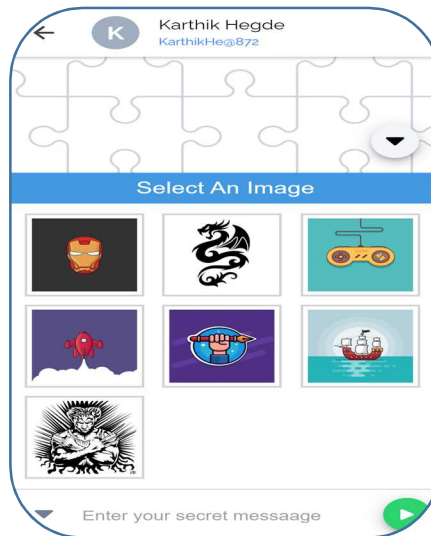
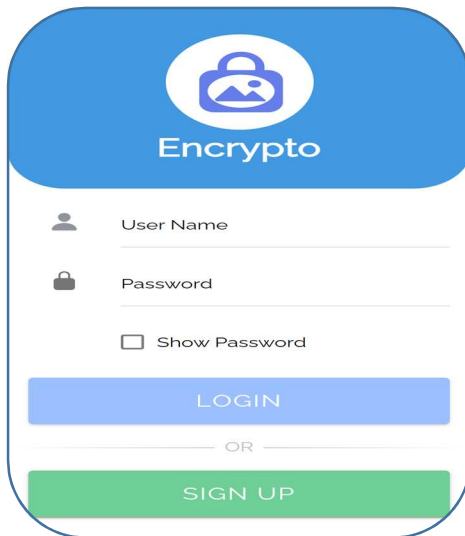
#### 4.1. Module Description

- i. **Registration:** The user fills out the details such as name, password, etc. to register into the system.
- ii. **Login:** In order to access the application feature, users need to fill in the proper username and password.
- iii. **Change Password:** This module is used to change the current password of the user
- iv. **Delete Account:** This module is used to delete the user account.
- v. **Encryption:** The image encryption is done here where image and text are provided
- vi. **Decryption:** The image is decrypted here in order to retrieve the data

#### 4.2. Tools, Language:

- **Tools:** Tools used are Android studio for converting web app to android.
- **Languages:** Languages used are html, scss, angular

### 4.3. Screenshots:



## 5. Testing

Login			
Test Case	Test Description	Test Output	Observed Output
1	If username is not entered	Login button is disabled	Successful
2	If password is not entered	Login button is disabled	Successful
3	If username or password not valid	Display Invalid username or password	Successful

Sign Up			
Test Case	Test Description	Test Output	Observed Output
1	If username is not entered	Login button is disabled	Successful
2	If password is not entered	Login button is disabled	Successful
3	If username or password not valid	Display Invalid username or password	Successful

Encryption			
Test Case	Test Description	Test Output	Observed Output
1	If message is not entered	Send button is disabled	Successful
2	If image is not selected	Send button is disabled	Successful



<b>Change Password</b>			
<b>Test Case</b>	<b>Test Description</b>	<b>Test Output</b>	<b>Observed Output</b>
1	If current password not valid	Display current password is not valid	Successful

<b>Delete Password</b>			
<b>Test Case</b>	<b>Test Description</b>	<b>Test Output</b>	<b>Observed Output</b>
1	If password not valid	Display invalid password	Successful

## 6. Conclusion and Future Enhancements

- **Conclusion:**

Through the development of the steganography android app, it is possible to send messages in the form of images which would help messages being intruded and getting hacked. Not only it provides a new way to transmit messages from sender to receiver without not being suspected by the hacker but provides a much safer way.

- **Future Enhancements:**

We could replace current algorithm with lsb steganography which would help to implement rsa algorithm over text before encrypting over it.

## 7. Reference

- <https://ionicframework.com/docs>
- [https://www.researchgate.net/publication/252024904\\_Image\\_steganographic\\_techniques\\_with\\_improved\\_embedding\\_capacity\\_and\\_robustness](https://www.researchgate.net/publication/252024904_Image_steganographic_techniques_with_improved_embedding_capacity_and_robustness)
- <https://codepen.io/caleboleary/pen/NNYoLW>